

Claims

- [c1] 1. A method for dynamically mapping Dynamic Multi-sourced Persisted EJB attributes to source system resources, comprising:
- creating a context definition containing attributes representing collections of source system data;
 - specifying in an attribute mapping element whether each attribute in the context definition is mapped to a field in a data source;
 - storing the context definition in a persistent data cache;
 - creating an instance of a Dynamic Multi-sourced Persisted EJB;
 - applying the attributes in the context definition to the created instance of the Dynamic Multi-sourced Persisted EJB for mapping the specified attributes to source system data fields;
 - accessing mapped source system data by the Dynamic Multi-sourced Persisted EJB instance without requiring EJB compilation and deployment;
 - and
 - sending mapped attribute data from source systems to clients and from clients to source systems in response to client queries.
- [c2] 2. The method of claim 1, further comprising:
- identifying a data source system table where the attribute value is located if the attribute is mapped;
 - specifying whether each attribute in the context definition is cached; and
 - specifying access security requirements for each attribute in the context definition.
- [c3] 3. The method of claim 1, further comprising reloading the context definition during execution on demand.
- [c4] 4. The method of claim 1, further comprising reloading the context definition during execution on a schedule.
- [c5] 5. The method of claim 1, wherein the step of storing the context definition comprises creating a persistent cache table for managing context attributes during EJB Bean Managed Persistence lifecycle.

- [c6] 6. The method of claim 1, further comprising defining a key attribute to enable accessing mapped source system data through adapters.
- [c7] 7. The method of claim 1, further comprising specifying whether each mapped attribute in the context definition is mapped to a single field in a data source.
- [c8] 8. The method of claim 1, further comprising:
designating an attribute in the context definition to be mapped as a primary field in a data source; and
writing data from the primary field in the data source to other multiple mapped secondary fields in a data source.
- [c9] 9. The method of claim 1, further comprising creating, reading, updating, and deleting data from clients to source systems, and from source systems to clients, and updating attribute mapping elements.
- [c10] 10. The method of claim 9, further comprising keeping data in the cache synchronized and updated with the most recent data from clients to source systems, and from source systems to clients.
- [c11] 11. The method of claim 1, further comprising:
creating and loading a new context definition containing new attributes;
applying the new attributes to the Dynamic Multi-sourced Persisted EJB instance for mapping the new attributes to source system data fields during runtime;
recreating the persistent data cache; and
immediately sending new attribute data to clients.
- [c12] 12. The method of claim 1, further comprising representing the context definition as an XML document.
- [c13] 13. The method of claim 1, further comprising storing selected source and client data in the persistent data cache.
- [c14] 14. The method of claim 1, wherein the step of creating an instance of a Dynamic Multi-sourced Persisted EJB comprises creating and accessing an instance of a Dynamic Multi-sourced Persisted EJB from an external application

using generic method calls of an application programming interface selected from the group consisting of create(), find(), getAttr(), getAttrs(), getGuid(), setAttr(), setAttrs() and retrieveNewAndDeletedContexts().

- [c15] 15. The method of claim 14, further comprising performing runtime checks prior to executing a method call including querying a security engine to determine if the method call is authorized and querying back-end adapters to determine if there are pending back-end mapped data updates for keeping cache data synchronized and updated with back-end mapped data.
- [c16] 16. The method of claim 1, wherein the step of creating an instance of a Dynamic Multi-sourced Persisted EJB comprises creating and accessing an instance of a Dynamic Multi-sourced Persisted EJB from an external application through a Session EJB Wrapper using traditional method calls of an application programming interface selected from the group consisting of create(), getAttributeName() and setAttributeName().
- [c17] 17. The method of claim 16, further comprising performing runtime checks prior to executing a method call including querying a security engine to determine if the method call is authorized and querying back-end adapters to determine if there are pending back-end mapped data updates for keeping cache data synchronized and updated with back-end mapped data.
- [c18] 18. The method of claim 1, wherein the step of creating a context definition further comprises creating a Map/Cache/Secure table.
- [c19] 19. The method of claim 1, wherein the step of sending attribute data to clients comprises sending attribute data to client applications running on web browsers and sending attribute data to trusted Java applications running on client machines.
- [c20] 20. A computer-readable medium containing instructions for controlling a computer system to implement the method of claim 1.
- [c21] 21. A system for dynamically mapping Dynamic Multi-sourced Persisted EJB attributes to source system resources, comprising:

running on web browsers and sending attribute data to trusted Java applications running on client machines.

[c34] 34. A system for dynamically mapping Dynamic Multi-sourced Persisted EJB attributes to source system resources, comprising:

an application server including contexts connected to JMS adapters;
a data cache connected to the contexts in the application server for providing BMP data for mapping Dynamic Multi-sourced Persisted EJB attributes to back-end system data fields;
system adapters for connecting JMS adapters to back-end systems; and
an XML data storage device for providing context definition documents to the contexts and JMS adapters in the application server and to the system adapters.

[c35] 35. The system of claim 34, wherein the contexts include Dynamic Multi-sourced Persisted EJB instances and Session EJB Wrappers mapped to source system data.

[c36] 36. A system for dynamically mapping Dynamic Multi-sourced Persisted EJB attributes to source system resources, comprising:

- a context definition containing attributes representing collections of source system data;
- an attribute mapping element for specifying whether each attribute in the context definition is mapped to a field in a data source;
- the context definition being stored in a persistent data cache;
- an instance of a Dynamic Multi-sourced Persisted EJB being created;
- the attributes in the context definition being applied to the created instance of the Dynamic Multi-sourced Persisted EJB for mapping the specified attributes to source system data fields;
- mapped source system data being accessed by the Dynamic Multi-sourced Persisted EJB instance without requiring EJB compilation and deployment; and
- mapped attribute data being sent from source systems to clients and from clients to source systems in response to client queries.